NASA LANGLEY RESEARCH CENTER'S DISTRIBUTED MASS STORAGE SYSTEM

Juliet Z. Pao
D. Creig Humes

MS157A
NASA/Langley Research Center
Hampton, VA. 23681

Abstract

There is a trend in institutions with high performance computing and data management requirements to explore mass storage systems with peripherals directly attached to a high speed network. The Distributed Mass Storage System (DMSS) Project at the NASA Langley Research Center (LaRC) is building such a system and expects to put it into production use by the end of 1993. This paper presents the design of the DMSS, some experiences in its development and use, and a performance analysis of its capabilities. The special features of this system are: 1) workstation class file servers running UniTree software; 2) third party I/O; 3) HIPPI network; 4) HIPPI/IPI3 disk array systems; 5) Storage Technology Corporation (STK) ACS 4400 automatic cartridge system; 6) CRAY Research Incorporated (CRI) CRAY Y-MP and CRAY-2 clients; 7) file server redundancy provision; and 8) a transition mechanism from the existent mass storage system to the DMSS.

1. Introduction

The Distributed Mass Storage System (DMSS) project at the NASA Langley Research Center (LaRC) integrates emerging technologies from the areas of data storage hardware, high speed communications, and mass storage system software into a system that overcomes the limitations of the current approach to mass storage. The DMSS is characterized by peripherals attached directly to a network, and a workstation acting as the file server. The file server will no longer be an active participant in most data transfers because they will occur directly between the peripheral and the requesting client.

The first phase is a prototype system to provide a proof of concept. It will also provide a base for testing ideas, and measuring and tuning performance. Once the prototype system is successfully completed, the production phase of the project will be initiated. This phase will include the procurement of necessary production storage and the addition of other functionality, such as network-attached tape.

2. Background

The Analysis and Computational Division (ACD) is responsible for providing a Mass Storage System (MSS) to meet the storage needs for both central and distributed computing systems at the NASA LaRC. The current production MSS is implemented on LaRC's CRAY Y-MP. The system consists of a CRAY disk and three STK 4400 robotic tape libraries. The disk is managed by CRI's Data Migration Facility (DMF) software. When it fills to a site specified threshold, the DMF automatically moves selected files to the STK libraries. Files that reside on tape are transparently moved back to disk upon access.

The main access method to the MSS is through a set of LaRC-developed Explicit Archive and Retrieval System (EARS) commands (masput, masget, masls, etc.) which allow the users to put, get, list, move, remove, make and remove directories, and change attributes of MSS files. Files are transferred over the local area network to and from the CRAY disk. Users may also use the File Transfer Protocol (FTP) which is available for most network-attached machines.

The current MSS is typical of large scale mass storage systems in use today. Each transfer results in data flowing through the file server before arriving at its destination. In order to meet high performance demands, this server is

usually a supercomputer or mini-supercomputer. Because of the high cost of this class of machine, the current system has limited expandability, scalability, performance, and availability.

3. Goals

The primary goal of the DMSS project is to move away from costly proprietary hardware and software solutions towards an open systems approach that does not limit expandability or scalability. The hardware and software purchased and developed for the DMSS must adhere to industry standards. This will facilitate expandability, scalability, and changes to hardware and software platforms. Software used and developed must be portable so that LaRC efforts and experiences can benefit other sites with common mass storage requirements. The system must be capable of providing high-speed access to files for selected client machines (i.e. the supercomputers), while not penalizing the performance of other clients.

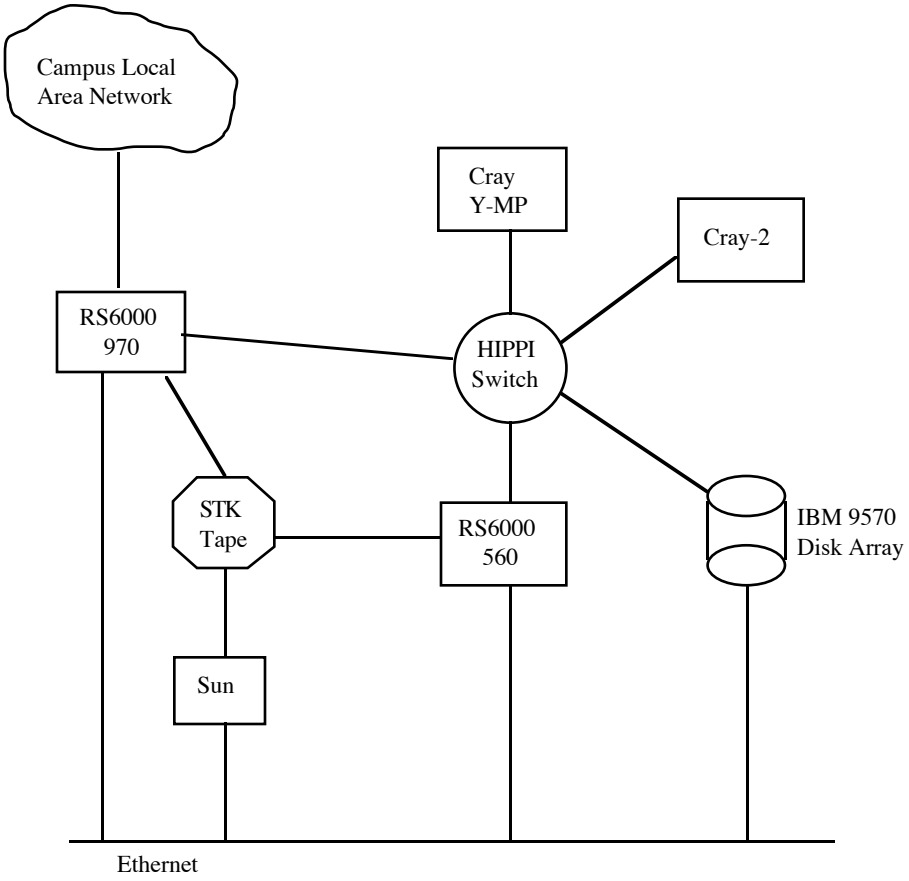

Figure 1

DMSS Prototype

4. DMSS Prototype

4.1 Equipment

The DMSS prototype [Figure 1] consists of an International Business Machines Corporation (IBM) 9570 disk array, two IBM RS6000 workstations (models 560 and 970), a CRAY Y-MP, and a CRAY-2. All of these pieces are connected to a Network Systems Corporation (NSC) PS32 High Performance Parallel Interface (HIPPI) Switch

[1,3]. The workstations are also connected to the existing STK 4400 tape libraries through a SCSI interface. A separate ethernet network connects the workstations and the disk array. This ethernet is used for disk array control and tape mount requests to the STK Sun workstation.

The disk array uses the Intelligent Peripheral Interface (IPI3) protocol [4]. IPI3 commands may be submitted to the disk array via either the HIPPI interface (using HIPPI/IPI3) or the ethernet interface. Data can be directed to flow through either interface. The current disk array supports the Redundant Array of Inexpensive Disks (RAID) level 3 and supplies 40 GB of storage.

The file servers for the prototype system are IBM RS6000s. Each file server currently has 3.5 GB of local disk, 128 MB of memory, and HIPPI and ethernet connections.

The CRAY supercomputers act as clients in the DMSS prototype system. They request data transfers from the file servers. The CRAY-2 has one HIPPI channel and the CRAY Y-MP has two.

The PS32 HIPPI Switch allows up to 32 machines or peripherals to be connected. The switch allows multiple HIPPI connections without any degradation to standard HIPPI performance. Switches may be hooked together to provide more connections.

UniTree, a product of OpenVision, is a mass storage system software package which manages a storage hierarchy for files. UniTree is available on almost all open system platforms. We are currently running version 1.0 of the National Storage Laboratory (NSL) UniTree. The NSL modified version 1.7 of the general UniTree product and made numerous enhancements. The enhancements of particular interest to the DMSS project are support for HIPPI-attached disk arrays and multiple dynamic storage hierarchies. UniTree provides FTP and NFS interfaces to its filesystem and also supports distributing pieces of the system to different machines (i.e. one machine can support tape functions while another supports the disk cache).

4.2 Data Flow in the DMSS

Throughout the rest of this paper, components of the DMSS will be discussed in terms of the IEEE Mass Storage Reference Model (MSRM), Version 4, and the current evolution of Version 5 [5,7].

Clients of the DMSS that have HIPPI channels and the appropriate software drivers can take advantage of the speed of the disk array. These machines have bitfile client software which sends UniTree file transfer requests to the file server. UniTree then instructs the disk array to transfer data to/from the HIPPI port specified in the file transfer request. The disk array then initiates the data transfer with the requesting client's software component, called the mover, which moves data between the proper memory address and the HIPPI channel. The protocol used to accomplish the data transfer is IPI3 third-party [8].

Other clients of the DMSS, which do not possess HIPPI channels, cannot trade data directly with the disk array. For these clients, one of the file servers acts as an intermediary. The file server receives requests from them through a standard protocol (FTP or RCP). The file server then transfers data between the client (through FTP or RCP) and disk array (through IPI3 third party). It is worth noting here, while hundreds of these clients exist and make use of the current MSS, they only account for approximately twenty percent of all data transferred.

The STK libraries are connected to the file servers and do not have HIPPI connectivity. During a file migration, a file server acts as a HIPPI client (as described above) to get data from the disk array before it writes the data to the tape. During a file recall a file server reads the data from tape before sending it to the disk array.

The initial user interfaces supported by DMSS include FTP, RCP, and EARS. All of these interfaces are explicit file transfer mechanisms which transfer complete files sequentially.

4.3 Redundancy

The approach for providing high availability is through redundant equipment. The production system will consist of two disk arrays, two workstations, and two HIPPI switches. This allows for the loss of any single piece of equipment without incurring lengthy down time. There are external SCSI disks that house the NSL/UniTree

databases. Upon the loss of one server the other can be reconfigured to take over the functionality of the unavailable server, with access to the most up to date databases. The redundancy of equipment also allows for new system testing and development without impacting production use.

5. Prototype Development Work

The prototype system required LaRC to undertake development and integration work. The areas that needed development were IPI3 third party movers for the CRAY machines, user interfaces, and a mechanism to transition our current production system data to DMSS in an efficient manner.

5.1 Mover for the CRAY Y-MP with Model E Input/Output Subsystem (IOS)

In order to provide third-party transfer for the supercomputer client, movers have been developed for both user space and kernel space. The kernel version has been chosen for production use because it allows access to DMSS from multiple processes and fair sharing of the mover's system resource, the HIPPI channels. The user space version only allows one process to access the HIPPI channel at a time.

Mover Interface

The bitfile client, which is a set of NSL UniTree functions, communicates with both UniTree and the mover. It communicates with the mover by issuing transactions which consist of the following information:

function - action to be performed (such as read, write, or cancel)
transaction identifier - a 32-bit integer which uniquely identifies the transaction
buffer - a pointer to a buffer
length - the data length in bytes of the transaction
device index - the device index of the HIPPI device used for this transaction
status - pointer to a status structure associated with this transaction

When the bitfile client issues a transaction to the mover, it also issues a companion request to the file server which results in the file server issuing one or more IPI3 third-party transfer requests to the disk array system. The disk array system then sends the waiting client's mover one or more Transfer Notification Responses (TNR), each of which contains a Transfer Notification Parameter (TNP) with the following information:

transaction identifier- a 32-bit integer which uniquely identifies the transaction
offset- offset in bytes of this segment relative to the beginning of the transaction
length- data length in bytes of this segment
last_transfer_flag - flag to indicate that this request is the last transfer for the transaction identifier

The mover uses the TNP information to take action to complete the third-party transfer. One transaction request from the UniTree bitfile client may result in multiple TNRs due to file segmentation and system resource sharing requirements. The mover makes no assumptions as to the order of arrival or segment length of these TNRs. It also does not assume that all TNRs for a particular transaction identifier must arrive before it can handle the TNR of another transaction identifier. [8]

Mover Design

The mover maintains transaction queues and other information necessary to manage requests from multiple processes. The mover also maintains two kinds of internal buffers. It owns three large buffers used to receive the TNR and data, and many small ones used to store the HIPPI-FP (Framing Protocol) header and IPI3 command for a write request. The buffers are necessary because the mover must always be ready to accept a TNR for any transaction in the system.

The size of the large buffer limits the amount of input data coming from the disk array system via UniTree. As the buffer size increases, the number of HIPPI packets needed to perform the transfer decreases. An appropriate buffer size must be chosen to maximize performance and minimize waste of memory. The raw HIPPI driver on the CRAY Y-MP can handle a HIPPI write that has data split between two buffers. Therefore, the mover only needs to provide

small buffers for the HIPPI-FP header and IPI3 command, and the user data does not need to pass through an intermediate buffer on a write. The size of the output packet is slightly larger than the user buffer size and is only limited by the maximum size of a HIPPI packet supported by the Model E IOS.

There is a set of commands to provide the following operational capabilities for the control of the mover:

- Initialize the mover environment.
- Halt all mover operations immediately (without shutting down the supercomputer client).
- Disable the submittal of transactions.
- Drop all active transactions.
- Close all HIPPI devices.
- Clear mover internal tables.
- Disable the submittal of transactions; all current transactions will be allowed to complete.
- Re-enable the submittal of transactions.
- Provide dynamic configuration capability for message logging options.
- Provide dynamic configuration capabilities for changing the time interval length for a transaction to be considered as timed-out and the time interval length to do the periodic checking.

5.2 Mover for the CRAY-2

The mover for the CRAY-2 is similar to that of the CRAY Y-MP, except for the handling of the third-party write. The raw HIPPI driver does not support a two buffer write. As a result, the mover's large buffers are used to pack the HIPPI-FP header, the IPI3 write command, and data into one contiguous area to be sent out with one HIPPI packet to the disk array system. So the bitfile client on the CRAY-2 can only submit requests to UniTree for transfers of size equal to or less than the large buffer size., Currently, the user space mover for the CRAY Y-MP has been ported to the CRAY-2. The porting of the kernel code began in June, 1993.

5.3 User Interfaces

The EARS commands have been rewritten for DMSS clients with HIPPI channels. These commands submit requests to NSL/UniTree using the supplied libnsl library. This library acts as the bitfile client and uses the LaRC developed mover for data transfer. This version of EARS is supported on the CRAY Y-MP, CRAY-2 and IBM RS6000.

Non-HIPPI attached machines have to retrieve their files from one of the file servers. These machines can get data either through FTP, RCP, or EARS. FTP is provided with UniTree. Two options are currently under investigation for providing RCP access. The first uses a locally modified version of RCP that understands how to talk to UniTree and the disk array (much like the EARS commands for the CRAYs). The second is to NFS mount the UniTree file system and use the regular RCP. The modified RCP currently works, but NFS with the disk array does not, so no comparison of performance is available at this time. The EARS interface is available to all distributed machines and is built using RCP for file transfers.

5.4 Transitioning From the Present DMF/UNICOS System to NSL/UniTree

The current LaRC MSS has more than a million files which comprise 1.5 terabytes of data on the STK ACS 4400 tape library under DMF management. LaRC has developed software that provides a mechanism for users to access any data in the current mass storage system on the first day of DMSS usage. The transition of DMF data into the DMSS is transparent to the users and requires minimal down time for the current system.

The day before DMSS production, the current mass storage system will be shut down for the transition process to take effect. First, on the CRAY Y-MP, a database called LaRCDB will be created using inode information of the current mass storage file system, the DMF daemon database, and the tape catalog database. The LaRCDB will then be moved to the file server. For each entry in LaRCDB, an entry will be created in the UniTree name server with a special flag set, indicating that it is a DMF formatted file. When a DMF file is accessed by a user via UniTree, the DMF flag will result in the tape file being staged onto UniTree disks using locally-developed routines incorporated into UniTree. After the staging, the DMF file becomes a bona fide UniTree file and its entry in the LaRCDB will be marked as soft-deleted.

While all the DMF files are available for UniTree users when they access them, not all of those files will be accessed by the users. So after DMSS is in production, a utility will be run on non-prime shifts to transition DMF files, cartridge by cartridge, into bona fide UniTree files until all files have been transferred.

6. Current Status

The prototype system is currently in a functional state. Test files are constantly being transferred, compared, and migrated. A majority of the effort now is spent testing and stabilizing the locally developed software and NSL/UniTree. The major items still in development are the CRAY-2 kernel mover and the transition software.

6.1 Performance of the DMSS

The initial tests of accessing DMSS data on the disk array system have been encouraging. The performance figures are grouped into three parts: disk array performance, file transfer performance to and from the CRAY Y-MP with Model E IOS, and file transfer performance between a Sun workstation and DMSS. The Sun is connected to the local area network via ethernet. The supercomputer's statistics were gathered on an idle machine, whereas the statistics for the local area network access were gathered in a normal production traffic environment. The IBM 9570 disk array system is configured using a 64K block size. All file transfer performance measurements include the whole transfer time between the client disk and the UniTree-managed disk array.

Disk Array Performance

Figure 2 shows the performance for the IBM 9570 disk array in both the first-party and third-party modes. Third-party performance was gathered using the CRAY Y-MP as the client and the IBM RS6000 560 as the file server. The performance includes the overhead of the command and response packets sent over the ethernet for control.

Complete File Transfer Between CRAY Y-MP and the DMSS

The timing measured is for file sizes of .5MB, 2MB, 16MB and 64MB, which are all block-aligned. Transfers that are block-aligned occur directly between the disk array and the CRAY. For non-aligned parts of a transfer, the file server is responsible for performing the transfer with the disk array [8]. In this case, the file server gets data from the CRAY's mover and places it on the disk array. This part of the transfer has been observed to take between 0.06 and 0.5 seconds.

Figure 3 compares the DMSS read transfer rates of different file sizes using large buffer sizes of 1MB, 2MB and 4MB. The graph for the 4 MB buffer case shows a decrease of performance as the file size increases from 16MB up to 64MB. This is due to the time necessary to flush the CRAY disk cache buffer. The performance of the current system is also plotted to show the increase of performance of DMSS.

Figure 4 compares the DMSS write transfer rates of different file sizes using large buffer sizes of 1MB, 2MB and 4MB. The write scenario is not limited by the large buffer size but rather the user level program's, namely masput's, buffer size. The graph shows that changing the user level buffer size from 2MB to 4MB did not yield a proportional increase of performance. The performance of the current system is also plotted for comparison. The CRAY's disk buffer cache was cleared before each transfer.

Figure 2 shows that larger buffers give increasingly better results. This is true for data transfers between the disk array system and the client's memory, but not for disks to disk file transfers. Both Figures 3 and 4 support the choice of 2MB for the mover's internal large buffer and user level program's buffer. Choosing buffer sizes larger than this gives rapidly diminishing returns due to the CRAY disk speed and the size of the CRAY disk buffer cache.

Complete File Transfer Between the LaRC Local Area Network and the DMSS

Figure 5 gives the statistics for DMSS access from a Sun workstation on the LaRC campus local area network. Masput and masget make use of the modified RCP (on the file server) which talks directly to UniTree. The performance of the current system is also plotted for comparison.

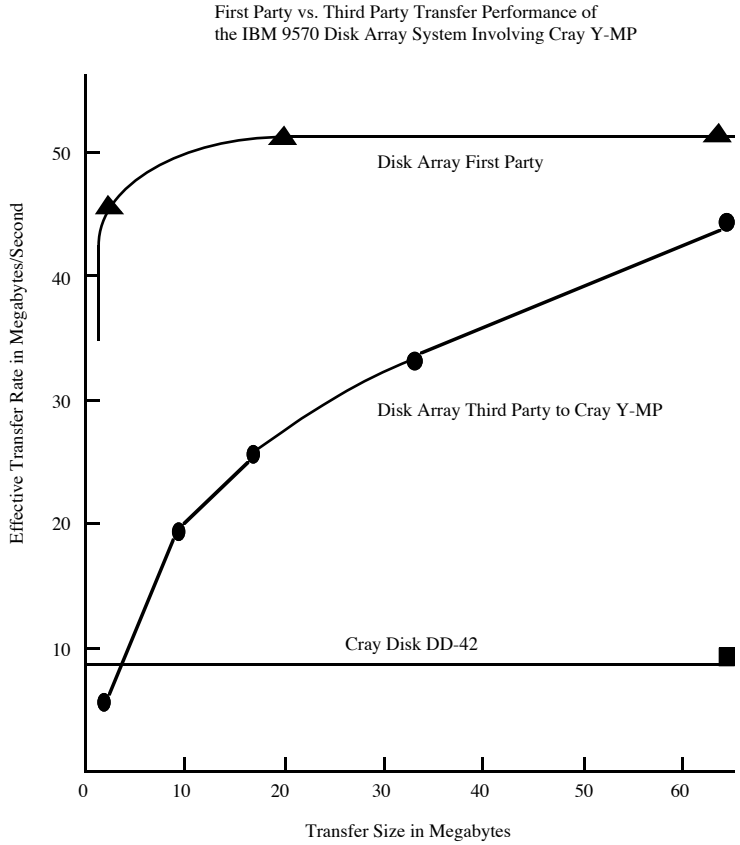

Figure2. Performance comparison among the first party disk array transfer rates provided by IBM, the third party disk array transfer to/from Cray Y-MP using the LaRC mover, and the sustained transfer rate of the Cray DD-42 disks.

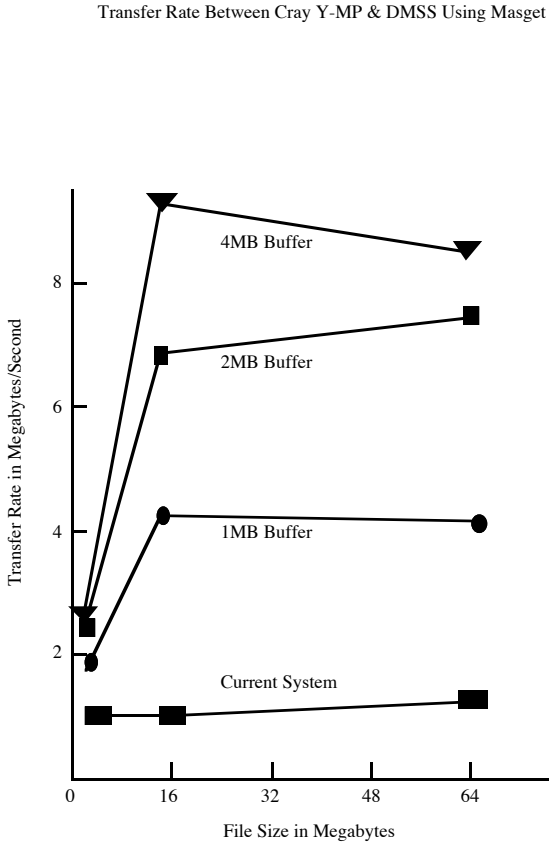

Figure 3. Transfer rate comparison of masget using different sizes of buffers on the Cray Y-MP.

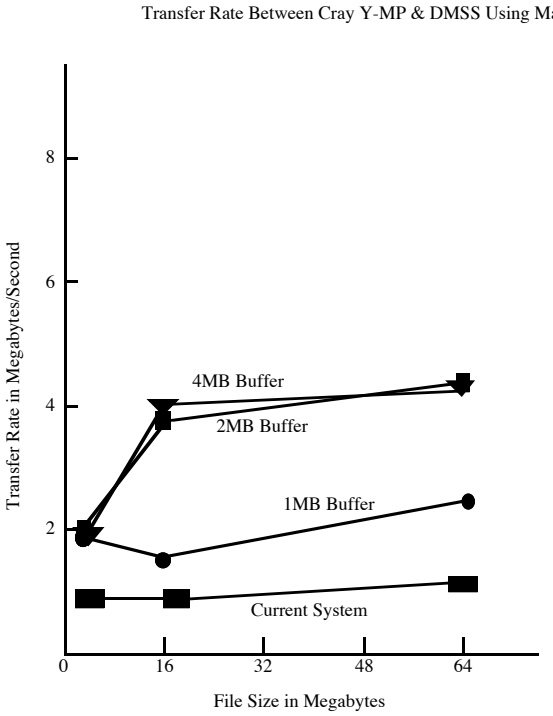

Figure 4. Transfer rate comparison of masput using different buffer sizes on the Cray Y-MP.

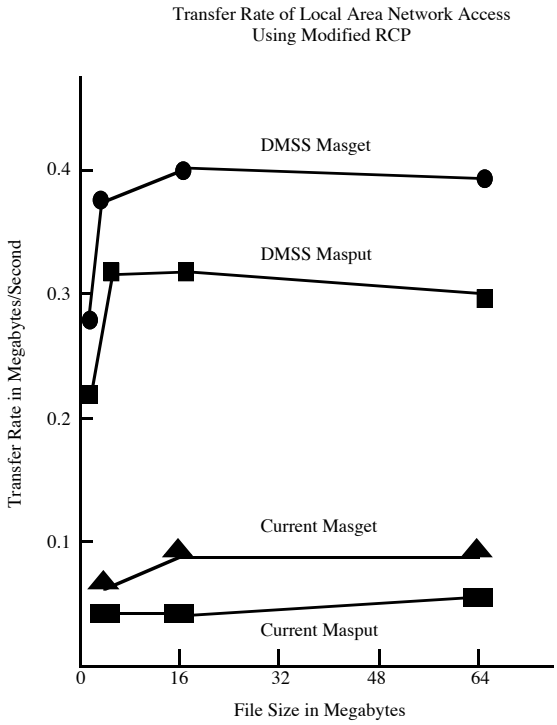

Figure 5. Transfer rate comparison of masput and masget used from machines on the LaRC local area network.

6.2 Schedule

Development will continue through the summer of 1993, along with debugging efforts for existing components and NSL/UniTree. Internal test users will begin making use of the system sometime in August and will use the system for a two month evaluation period. If the system is stable at this point selected users from the research community will be invited for a one to two month beta-test, followed by full production use by the entire research center. A second 40 GB HIPPI-attached disk array, external SCSI disk, and second HIPPI switch will be added to the configuration before production usage is initiated.

7. Future Plans

Once DMSS is stable, other features will be added. Of particular interest is a file system interface (using vnodes). The first supported interfaces are all disk-to-disk file transfers. There is also a need for high performance data transfers directly between an application on the CRAYs and the disk array. Currently the only way to do this is to incorporate the libnsl routines directly into a program. This does not give the users file location transparency, thus placing an unnecessary burden on the users. A transparent file system interface would allow for extremely good performance for jobs running on the CRAYs, while maintaining location transparency. In this way all permanent file storage for the CRAYs can be managed by DMSS.

Also of interest is a site-wide distributed file system that will be able to use the DMSS to store data. For example, this could be based on OSF's DCE/DFS.

Other machines with HIPPI attachments will have movers developed to enable high speed DMSS access. The next machine targeted is the Intel Paragon.

LaRC will also pursue adding network-attached tape to DMSS. This will relieve the workstations of more than 95 percent of the data transfer responsibilities of the current CRAY Y-MP based MSS. Migrations and recalls will occur directly between network peripherals. As the multiple dynamic hierarchies mature, applications, such as backup and visualization, will move data directly to and from the network-attached tape.

8. Conclusion

When DMSS goes into production in the fall of 1993, it will relieve the CRAY Y-MP of its function as a file server. Users of DMSS will experience performance three times better than the current system. Their access to DMSS will no longer be interrupted by the file server's unavailability due to various system maintenance functions, malfunctions, or system time. The system will be expandable and scalable. Disk and tape will be added directly to the network as the need grows. If one file server is not powerful enough to handle the workload, then the function can be split among two or more file servers.

9. Acknowledgments

The LaRC prototype DMSS system has gone through the cycle of design, acquisition, testing and software development since January 1991. The acquisition took the initial one and a half years. We would like to acknowledge Everett C. Johnson and David E. Cordner of the Computer System Branch at NASA LaRC for their help in the design and acquisition of DMSS equipment, the Unisys Cooperation for their support in software development and testing, and CRAY Research Inc. for their support on the UNICOS internals. We also appreciate the cooperation of DISCOS of General Atomics (presently OpenVision) and IBM Federal Systems Company.

References

1. ANSI, "High Performance Parallel Interface - Mechanical, Electrical, and Signaling Protocol Specification (HIPPI-PH)", American National Standards Institute, X3.183-1991.

2. ANSI, "High Performance Parallel Interface - Framing Protocol (HIPPI-FP) Preliminary Draft", American National Standards Institute, X3.210-199x.

3. ANSI, "High Performance Parallel Interface - Physical Switch Control (HIPPI-SC)", American National Standards Institute,X3.91-023-1991.

4. ISO/IEC, "Information Technology - Intelligent Peripheral Interface Part 3: Device Generic Command Set for Magnetic and Optical Disk Drives", ISO/IEC 9318-3, September, 1990.

5 Coleman, S. and S. Miller, eds., "Mass Storage System Reference Model Version 4", IEEE Technical Committee on Mass Storage Systems and Technology, May 1990.

6. Coyne, R. and H. Hulen, "An Introduction to the Mass Storage System Reference Model, Version 5", Proc. Twelfth IEEE Symposium on Mass Storage Systems, Monterey, April 1993.

7. Merrill, J., "Toward a Standard IEEE Mover", Proc. Twelfth IEEE Symposium on Mass Storage Systems, Monterey, April 1993.

8. Hyer, R., R. Ruef, R. Watson, "High-Performance Data Transfers Using Network-Attached Peripherals at the National Storage Laboratory", Proc. Twelfth IEEE Symposium on Mass Storage Systems, Monterey, April 1993.